

FEATURE RECOGNITION BERBASIS CORNER DETECTION DENGAN METODE FAST, SURF DAN FLANN TREE UNTUK IDENTIFIKASI LOGO PADA AUGMENTED REALITY MOBILE SYSTEM

(Feature Recognition Base on Corner Detection with FAST, SURF and FLANN TREE Method for Logo Identification in Augmented Reality System)

Rastri Prathivi

Program Studi Teknik Informatika, Jurusan Teknologi Informasi
Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang
Email: vivi@usm.ac.id

Abstract

Logo is a graphical symbol that is the identity of an organization, institution, or company. Logo is generally used to introduce to the public the existence of an organization, institution, or company. Through the existence of an agency logo can be seen by the public. Feature recognition is one of the processes that exist within an augmented reality system. One of uses augmented reality is able to recognize the identity of the logo through a camera. The first step to make a process of feature recognition is through the corner detection. Incorporation of several method such as FAST, SURF, and FLANN TREE for the feature detection process based corner detection feature matching up process, will have the better ability to detect the presence of a logo. Additionally when running the feature extraction process there are several issues that arise as scale invariant feature and rotation invariant feature. In this study the research object in the form of logo to the priority to make the process of feature recognition. FAST, SURF, and FLANN TREE method will detection logo with scale invariant feature and rotation invariant feature conditions. Obtained from this study will demonstration the accuracy from FAST, SURF, and FLANN TREE methods to solve the scale invariant and rotation invariant feature problems.

Keywords: *Feature Recognition, Logo Identification*

1. PENDAHULUAN

Logo adalah simbol berbentuk grafis yang merupakan identitas dari sebuah organisasi, lembaga atau perusahaan. Logo umumnya digunakan untuk memperkenalkan kepada *public* keberadaan sebuah organisasi, lembaga atau perusahaan. Logo juga menjadi simbol jaminan mutu bagi sebuah hasil produksi dan telah menjadi keharusan yang wajib dimiliki oleh sebuah organisasi, lembaga atau perusahaan. Keberadaan logo di dalam sebuah organisasi, lembaga atau perusahaan menjadi penting karena melalui logo maka ciri khas atau informasi

yang berkaitan dengan organisasi, lembaga atau perusahaan tertentu dapat diketahui. Informasi yang terkandung di dalam sebuah logo dapat disampaikan menggunakan media komunikasi seperti *mobile device* dan internet menggunakan aplikasi *augmented reality*. Untuk bisa mengenali bentuk sebuah logo di dalam sistem *augmented reality* maka terlebih dahulu perlu dilakukan proses *recognition* terhadap logo tersebut.

Berkaitan dengan jenis *image* yang dapat dideteksi di dalam proses *recognition*, maka ada dua jenis *image* yaitu *bitmap* dan *vector*. Logo termasuk ke dalam kategori *vector image*. Di

mana tingkat resolusi di dalam sebuah *image* berbentuk logo tidak memiliki peranan yang penting. Proses *recognition* yang diterapkan pada jenis *image* yang berbeda akan mempengaruhi keakuratan proses *recognition*. Proses *recognition* merupakan proses awal yang perlu ada untuk membangun sistem *augmented reality*. Proses *recognition* ini bertujuan untuk mengidentifikasi bentuk logo agar dapat dikenali, dideteksi dan dideskripsikan oleh sistem. Di dalam *recognition* yang berbasis *feature*, proses awal yang perlu dilakukan adalah melakukan pendeteksian *corner*. Pendeteksian *corner* akan mendeteksi bentuk sebuah logo yang berorientasi pada sudut dalam logo tersebut. Logo akan sulit dideteksi jika letak sudut saat dideteksi tidak sesuai dengan data yang ada.

Selain itu letak sudut akan berbeda jika ada perbedaan kondisi tampilan seperti perbedaan atau perubahan skala dan rotasi. Maka perlu adanya metode *corner recognition* yang tepat untuk mengatasi masalah tersebut. Proses *recognition* berbasis *corner feature* perlu mengalami tiga tahapan yaitu *feature detection* dan *extraction*, *feature descriptor* dan *feature matching*. Setiap tahapan tersebut dapat memiliki metode tertentu yang dapat dijadikan acuan.

Dalam *state of the art* tentang *recognition* berbasis *corner feature* untuk sistem *augmented reality* ada beberapa penelitian yang telah membuktikan penggunaan beberapa metode yang berbeda-beda antara lain: Mikolajczyk, K. dan Schmid, C. [1] di dalam *papernya* yang berjudul *Performance Evaluation of Local Descriptor* melakukan penelitian terhadap beberapa *feature descriptor* dan menemukan bahwa *GLOH* (*Gradient Location and Orientation Histogram*) *descriptor* yang merupakan variant dari *SIFT* (*Scale Invariant Feature Transformation*) *descriptor* memiliki komputasi yang tinggi meskipun dapat mengatasi masalah pada perbedaan skala lebih baik daripada *SIFT descriptor*. Kelemahan dari penelitian ini adalah belum dapat mengatasi masalah pada kondisi perbedaan rotasi di dalam sebuah *image* dan hanya melakukan penelitian terhadap *feature descriptor*.

Wei-Chao Chen, Yingen Xiong, Jiang Gao, Natasha Gelfand, dan Radek Grzeszczuk [10] di dalam *papernya* yang berjudul *Efficient Extraction of Robust Image Features on Mobile Devices* melakukan penelitian terhadap SURF untuk melakukan tahapan *feature extraction* dan *feature descriptor* serta ANN (*Aproximate Nearest Neighbor*) untuk melakukan tahapan *feature matching*. Penelitian ini menghasilkan komputasi rata – rata sebesar 30% lebih cepat dibandingkan dengan penelitian yang dilakukan oleh Herbert Bay, Tinne Tuytelaars, Luc Van Gool [3]. Obyek penelitian mereka adalah *image* berjenis bitmap atau foto.

Gabriel Takacs [9] di dalam *papernya* yang berjudul *Outdoors Augmented Reality on Mobile Device Using Loxel-Based Visual Feature Organization* membangun sistem *outdoor augmented reality* dengan memanfaatkan proses *image recognition* di sisi *client* dan proses *matching* di sisi *server*. Mereka menggunakan metode SURF, ANN dan Brute Force untuk melakukan proses *recognition* di sisi *client* pada *mobile device*. Obyek penelitian mereka adalah *real time video* yang diambil secara langsung.

Niels Henze, Torben Schinke dan Susanne Boll [8] di dalam *papernya* yang berjudul *What is That? Object Recognition from Natural Features on a Mobile Device* menggunakan penggabungan metode yang berbeda – beda untuk setiap tahapan proses pada *recognition*. Mereka menggunakan metode FAST, SIFT dan Vocabulary Tree. Obyek penelitian mereka adalah poster yang berukuran 45x55 cm. Proses *recognition* yang diterapkan berbasis *natural features* yaitu melakukan pendeteksian *image* sesuai dengan bentuk *imagennya*.

Daniel Wagner [5] dan teman-teman di dalam *papernya* yang berjudul *Real Time Detection and Tracking for Augmented reality on Mobile Device* menyatakan bahwa metode SIFT sangat kuat untuk melakukan proses *recognition* tetapi memerlukan komputasi yang tinggi. Mereka melakukan modifikasi pada metode SIFT sehingga dapat diimplementasikan pada *mobile device*. Mereka menggunakan metode FAST pada proses *interest point detection* dan ekstraksi *image*, proses *feature descriptor*

menggunakan metode *SIFT* dan proses *feature matching* menggunakan metode *Spill Forest*. Obyek penelitian mereka beragam yaitu *foto*, *panorama picture*, *image satellite*, dan *vector image*.

Charles Norona, Tyagi dan Vivek Kumar [7] di dalam papernya yang berjudul *Non-distributed Object Recognition Potential on the Android Platform* menerapkan proses *recognition* pada *mobile device* dengan platform android menggunakan metode *SURF*. Dari penelitian ini mereka membuktikan bahwa *SURF* lebih cepat melakukan komputasi proses *feature detection* dan *feature descriptor* pada platform android. Obyek penelitian mereka adalah *image photographic* yang diambil secara langsung dari kamera.

Mosalam Ebrahimi dan Walterio W. Mayol-Cuevas [6] di dalam papernya yang berjudul *Adaptive Sampling for Feature Detection, Tracking and Recognition on Mobile Platforms* memodifikasi metode *FAST (Features from Accelerated Segment Test)* untuk diimplementasikan ke dalam *handphone*. Hasil modifikasi dari metode *FAST* adalah *M-FAST (Mobile Features from Accelerated Segment Test)*. Selain itu mereka juga memodifikasi *BRIEF* untuk membentuk *adaptive descriptor*. Mereka menyatakan bahwa metode *M-FAST* yang mereka temukan lebih cepat dari metode *SURF*. Sedangkan untuk proses *feature matching* mereka menggunakan metode *FLANN TREE*. Obyek penelitian yang mereka teliti adalah video atau gambar bergerak.

Sami Mohammad Halawani, Dzulkifli Mohammad [20] di dalam papernya yang berjudul *Logo Matching Technique Based on Principle Component Analysis* melakukan penelitian terhadap logo dengan metode *principle component analysis*.

Dari *state of the art* tentang proses *recognition* maka tingkat akurasi dari penelitian yang berhubungan dengan *vector image* masih

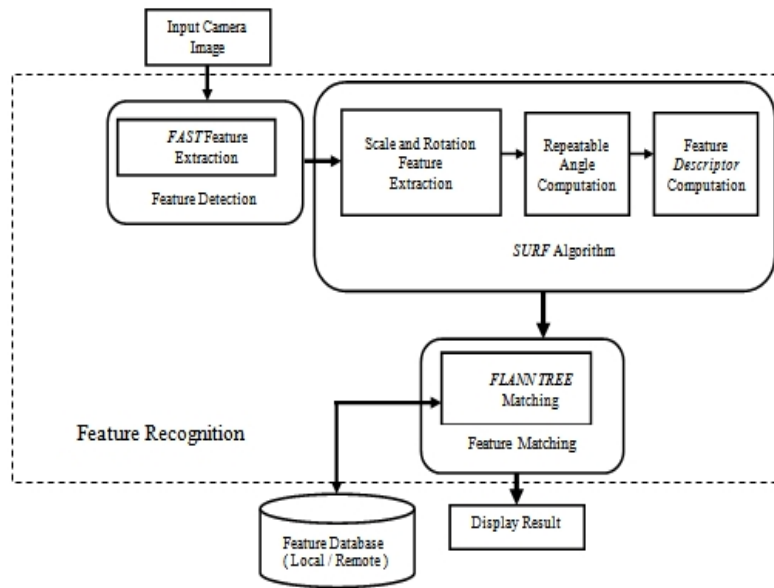
beragam meskipun tidak sepenuhnya menggunakan ketiga tahapan proses *recognition*. Sehingga belum ditemukan metode yang akurat untuk melakukan proses *recognition* dengan menggunakan ketiga tahapan proses *recognition* dengan metode yang berbeda di setiap tahapannya.

Proses *recognition* pada logo sebagai *vector image* sulit dilakukan jika ada perbedaan kondisi tampilan seperti *scale invariant* dan *rotation invariant*. Belum ditemukannya metode yang tepat dan tingkat akurasi yang tinggi dalam proses *feature recognition* pada logo berbasis *corner detection* dalam *augmented reality system* melalui tahapan *feature extraction*, *feature descriptor* dan *feature matching*.

Penelitian ini bertujuan menemukan metode yang tepat untuk mengatasi masalah dalam proses *recognition* logo pada kondisi *scale invariant* dan *rotation invariant*.

Meningkatkan akurasi pada proses *feature recognition* logo berbasis *corner detection* melalui tahapan *feature extraction*, *feature descriptor* dan *feature matching* dengan metode *FAST*, *SURF* dan *FLANN TREE*.

Gambar 1 adalah ilustrasi proses dari *feature recognition* berbasis *corner detection* dengan metode *Fast*, *Surf* dan *Flann Tree*. Logo akan *capture* melalui kamera. Ketika logo *capture* akan ada tiga kondisi yang diberikan yaitu normal, perbedaan skala (*scale invariant*), dan perbedaan rotasi (*rotation invariant*). Kemudian data dari logo *capture* akan melalui proses *corner detection* dan *extraction* dengan menggunakan metode *Fast*. Setelah diperoleh letak *corner* maka proses selanjutnya adalah membentuk *descriptor* dengan metode *Surf*. Dari pembentukan *descriptor* ini akan diperoleh data yang akan menunjukkan letak *corner* pada logo yang *capture* dengan data logo di dalam database. Kemudian proses *matching* bisa dilakukan dengan metode *Flann Tree*.



Gambar 1. Kerangka Teori Proses *Feature Recognition*

Yang terdiri dari:

- Input camera untuk mengambil image dari kamera perangkat mobile.
- Feature detection dilakukan dengan metode FAST
- Feature descriptor dengan metode SURF
- Feature matching menggunakan metode FLANN TREE
- Dan display result akan menampilkan hasil feature matching dari pengambilan image yang berupa logo dengan logo di dalam database lokal.

2. PEMBAHASAN

2.1. Pengolahan Awal Data

Pengolahan awal data dilakukan terhadap data berbentuk *image* yang berjenis logo sebanyak 50 jenis logo. Pengolahan awal data berbentuk logo ini terdiri dari beberapa tahap, yaitu:

- Menentukan resolusi untuk setiap logo yaitu dengan resolusi maksimal 320 x 320 pixel.
- Data yang berupa data sumber adalah data yang bersifat normal artinya tidak ada perlakuan khusus seperti perbedaan skala, perbedaan rotasi dan memiliki kualitas warna standar.
- Ketika data logo sebagai data sumber diinput ke dalam sistem maka warna pada logo akan diubah menjadi warna *grayscale*.

- Data kedua akan diinput melalui kamera pada sistem, data ini akan ditangkap oleh kamera sesuai dengan warna aslinya.
- Saat data kedua ditangkap oleh kamera, selain logo diperlakukan pada kondisi normal, ada kondisi khusus yang diberikan pada logo yaitu adanya perbedaan skala dan rotasi sebesar 90° dan 180° pada data logo.
- Kemudian sistem akan melakukan proses identifikasi logo pada data sumber dan data yang diinput melalui kamera.

2.2. Tahapan Proses Recognition

Tahapan proses untuk melakukan eksperimen logo *recognition* sebagai berikut:

Tahap 1: Proses *feature detection* dan *extraction* dengan metode FAST

- Penentuan intensitas *pixel*

- Pada tahap *feature detection* ditentukan *segment test* pada masing – masing logo dari 50 logo yang merupakan obyek penelitian.
- Intensitas pixel dari *segment test* yang diambil ditentukan dengan rumus (1).

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \text{ (darker)} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \text{ (similar)} \\ b, & I_p + t \leq I_{p \rightarrow x} \text{ (brighter)} \end{cases} \quad (1)$$

- Pada rumus di atas, dihitung untuk setiap pixel $p \in P$ (P adalah semua *pixel* dalam *image* yang ditraining).
- P dikelompokkan ke dalam tiga subset yaitu P_d, P_s, P_b dan setiap pixel p dikelompokkan ke dalam subset $P_{S_{p \rightarrow x}}$.

f. Penentuan dan perhitungan *corner*

- Setelah pixel p dikelompokkan ke dalam subset $P_{S_{p \rightarrow x}}$ kemudian variabel *boolean* K_p ditentukan bernilai benar (*true*) jika p adalah *corner* dan bernilai salah (*false*) jika bukan *corner* (*non corner*).
- Untuk menentukan sebuah *pixel* merupakan *corner* dihitung dengan *entropy* dari K_p pada rumus (2).

$$H(P) = (c + \bar{c}) \log_2 (c + \bar{c}) - c \log_2 c - \bar{c} \log_2 \bar{c} \quad (2)$$

Di mana $c = |\{p | K_p \text{ is true}\}|$ (jumlah *corners*)

dan $\bar{c} = |\{p | K_p \text{ is false}\}|$ (jumlah *non corners*)

- Jumlah *corner* dan *non corner* dihitung dengan rumus (3)

$$H(P) - H(P_d) - H(P_s) - H(P_b) \quad (3)$$

Pendeteksian *corner* yang berdekatan dengan *non-maximal suppression*

Menentukan *non-maximal suppression* dengan mencari nilai maksimum V untuk setiap *corner* yang berdekatan berdasarkan rumus (4).

$$V = \max \left(\sum_{x \in S_{\text{bright}}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{\text{dark}}} |I_p - I_{p \rightarrow x}| - t \right) \quad (4)$$

Dengan definisi dari S_{bright} dan S_{dark} sebagai berikut:

$$S_{\text{bright}} = \{x | I_{p \rightarrow x} \geq I_p + t\}$$

$$S_{\text{dark}} = \{x | I_{p \rightarrow x} \leq I_p - t\}$$

- Jika nilainya lebih tinggi dari V_{max} maka setiap *corner* yang berdekatan akan dihapus.

Tahap 2: Proses *feature descriptor* pada metode SURF

- Pemfilteran *corner* dengan *haar wavelet*.
 - Menentukan gradien pada arah $dx, dy, |dx|$ dan $|dy|$.
- g. Menentukan vektor *descriptor* SURF berdasarkan rumus (5).

$$v_{\text{subregion}} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \quad (5)$$

Tahap 3: Proses *feature matching* dengan metode FLANN TREE berdasarkan rumus (6)

$$\text{cost} = \frac{s + w_b b}{(s + w_b b)_{\text{opt}}} + w_m m \quad (6)$$

- Penentuan index pada pencarian *nearest neighbor* dari *corner* logo sumber dan *corner* logo *capture* berdasarkan parameter dari *KD Tree Index*.
- Menentukan korespondensi dengan NNDR (*Nearest Neighbor Distance Ratio*).
- Menemukan *homography* berdasarkan *inlier* dan *outlier* dari logo sumber dan logo *capture*.

2.3. Pengujian

Pengujian dilakukan dengan mengacu pada beberapa kondisi yaitu kondisi *normal*, *scale invariant*, *rotation 90°* dan *rotation 180°*. Pada kondisi *normal* logo yang *capture* memiliki kondisi yang sama dengan data logo pada *database* sumber. Sedangkan pada kondisi *scale invariant*, logo yang *capture* memiliki perbedaan skala dengan data logo sumber. Selain itu pada kondisi *scale invariant*, logo yang *capture* memiliki *struktur image* yang berbeda

dengan logo pada data sumber. Pada kondisi *rotation* 90° dan *rotation* 180°, logo yang *dicapture* akan diputar dengan sudut 90° dan 180°.

Metode yang diuji adalah metode *Fast, Surf* dan *Flann Tree*; metode *Fast, Sift* dan *Flann Tree* dan metode *Fast, Brief* dan *Flann Tree*. Dari hasil pengujian didapatkan akurasi dari metode *Fast, Surf* dan *Flann Tree* lebih tinggi dibandingkan dengan metode *Fast, Sift* dan *Flann Tree* serta *Fast, Brief* dan *Flann Tree*.

Pengujian akan menghitung kecepatan waktu komputasi dari beberapa parameter yang digunakan sebagai acuan yaitu:

- Feature detection (ms): digunakan untuk menghitung lama waktu dilakukannya proses corner detection.
- Descriptor extraction (ms): digunakan untuk menghitung lama waktu dilakukannya proses ekstraksi descriptor untuk menentukan arah vektor descriptor.
- Descriptor indexing (ms): digunakan untuk menghitung lama waktu dilakukannya proses indexing pada descriptor yaitu memberikan nilai index pada letak vektor descriptor.
- Descriptor matching (ms): digunakan untuk menghitung lama waktu dilakukannya proses descriptor matching pada data sumber dan data yang diambil melalui kamera.
- Detect outlier dan GUI (ms): digunakan untuk menghitung lama waktu dilakukannya proses mendeteksi outline dan GUI dari logo.

Pengujian dengan parameter untuk menentukan akurasi area *corner* yang ditemukan yaitu:

- *Min matched distance*: digunakan untuk menghitung banyaknya area minimal yang sama antara data sumber dan data yang diambil melalui kamera.
- *Max matched distance*: digunakan untuk menghitung banyaknya area maksimal yang sama antara data sumber dan data yang diambil melalui kamera.
- *Corner quantity*: digunakan untuk menghitung banyaknya jumlah *corner*.

2.4. Hasil Eksperimen

- Metode *Fast, Surf* dan *Flann Tree*

Tabel 1. Rata-Rata Kecepatan Komputasi pada Metode *Fast, Surf* dan *Flann Tree*

	Feature Detection (ms)				Descriptor Extraction (ms)				Descriptor Indexing (ms)				Descriptor Matching (ms)				Detect Outlier and GUI (ms)			
Rata - Rata Kecepatan Komputasi	3.92	3.9	4	4.28	44.5	38.26	36.22	53.24	6.86	5.72	5.4	8.34	8.5	8.44	8.68	8.62	81.8	86.38	83.92	79.14

- Metode *Fast, Brief* dan *Flann Tree*

Tabel 2. Rata-Rata Kecepatan Komputasi pada Metode *Fast, Brief* dan *Flann Tree*

	Feature Detection (ms)		Descriptor Extraction (ms)		Descriptor Indexing (ms)		Descriptor Matching (ms)		Detect Outlier and GUI (ms)	
Rata - Rata Kecepatan Komputasi	3.84	3.74	5.74	5.42	3.66	3.34	5.7	5.56	63.84	64.74

- Metode *Fast, Sift* dan *Flann Tree*

Tabel 3. Rata-Rata Kecepatan Komputasi pada Metode *Fast, Sift* dan *Flann Tree*

	Feature Detection (ms)		Descriptor Extraction (ms)		Descriptor Indexing (ms)		Descriptor Matching (ms)		Detect Outlier and GUI (ms)	
Rata - Rata Kecepatan	3.84	3.66	221.82	202.24	10.36	8.78	10.68	10.58	96.58	78.88

Tabel 1,2, dan 3 memperlihatkan rata-rata kecepatan komputasi dengan parameter feature detection, descriptor extraction, descriptor indexing, descriptor matching dan detect outlier and GUI.

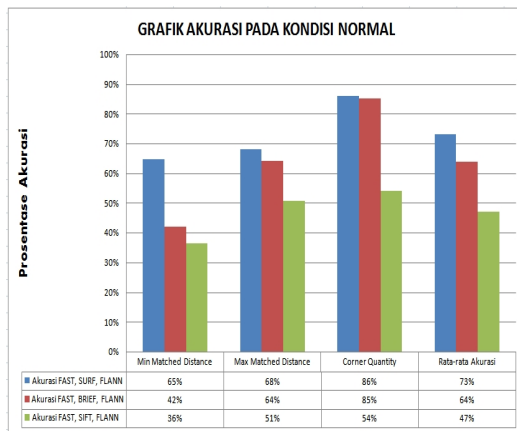
- Hasil Perbandingan Akurasi pada Kondisi Normal

Tabel 4. Perbandingan Akurasi pada Kondisi Normal

PERHITUNGAN / METODE	Min Matched Distance	Max Matched Distance	Corner Quantity	Rata-rata Akurasi
Akurasi FAST, SURF, FLANN	65%	68%	86%	73%
Akurasi FAST, BRIEF, FLANN	42%	64%	85%	64%
Akurasi FAST, SIFT, FLANN	36%	51%	54%	47%

Dari tabel 4 terlihat bahwa akurasi metode *Fast, Surf* dan *Flann Tree* memiliki tingkat rata – rata akurasi 9% lebih besar dibandingkan

metode Fast, Brief dan Flann Tree dan 26% lebih besar dibandingkan metode Fast, Sift dan Flann Tree.



Gambar 2. Grafik Akurasi pada Kondisi Normal

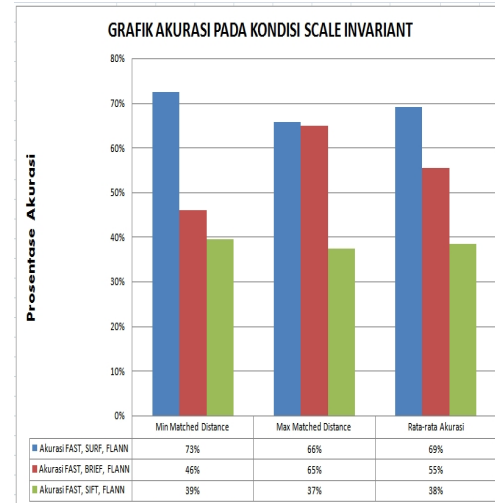
Dari gambar 2 diketahui bahwa akurasi metode Fast, Surf dan Flann Tree lebih tinggi dibandingkan dengan metode lainnya.

- Hasil Perbandingan Akurasi pada Kondisi Scale Invariant

Tabel 5. Perbandingan Akurasi pada Kondisi Scale Invariant

PERHITUNGAN / METODE	Min Matched Distance	Max Matched Distance	Rata-rata Akurasi
Akurasi FAST, SURF, FLANN	73%	66%	69%
Akurasi FAST, BRIEF, FLANN	46%	65%	55%
Akurasi FAST, SIFT, FLANN	39%	37%	38%

Dari tabel 5 terlihat bahwa akurasi metode Fast, Surf dan Flann Tree memiliki tingkat rata – rata akurasi 14% lebih tinggi dibandingkan metode Fast, Brief dan Flann Tree dan 31% lebih tinggi dibandingkan metode Fast, Sift dan Flann Tree.



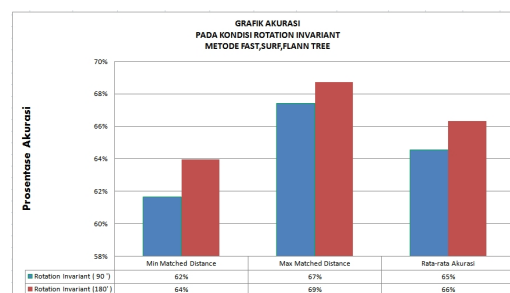
Gambar 3. Grafik Akurasi pada Kondisi Scale Invariant

Dari gambar 3 diketahui bahwa akurasi metode Fast, Surf dan Flann Tree pada kondisi scale invariant lebih tinggi dibandingkan dengan metode lainnya.

Tabel 6. Perbandingan Akurasi pada Kondisi Rotation Invariant

METODE FAST, SURF, FLANN TREE	Min Matched Distance	Max Matched Distance	Rata-rata Akurasi
Rotation Invariant (90°)	62%	67%	65%
Rotation Invariant (180°)	64%	69%	66%

Dari tabel 6 terlihat bahwa akurasi metode Fast, Surf dan Flann Tree memiliki tingkat rata – rata akurasi 65% pada sudut 90° dan 66% pada sudut 180°.



Gambar 4. Grafik Akurasi pada Kondisi Rotation Invariant

Dari gambar 4 diketahui bahwa memiliki rata-rata akurasi metode Fast, Surf dan Flann Tree pada kondisi *rotation invariant* 180° lebih tinggi dibandingkan dengan pada kondisi *rotation invariant* 90°, yang ditunjukkan oleh bar berwarna merah.

3. KESIMPULAN

Dari hasil penelitian terhadap logo *recognition* diperoleh kesimpulan sebagai berikut:

- a. Metode *Fast*, *Surf* dan *Flann Tree* akurat untuk melakukan identifikasi logo pada proses *feature recognition* berbasis *corner detection* yang terdapat di dalam sistem *augmented reality*.
- b. Metode *Fast*, *Surf* dan *Flann Tree* memiliki akurasi rata-rata 73 % untuk kondisi normal, 69% untuk kondisi *scale invariant*, 65% untuk kondisi *rotation* 90° dan 66% untuk kondisi *rotation* 180°.
- c. Akurasi metode *Fast*, *Surf* dan *Flann Tree* memiliki tingkat rata – rata akurasi 14% lebih tinggi dibandingkan metode *Fast*, *Brief* dan *Flann Tree* dan 31% lebih tinggi dibandingkan metode *Fast*, *Sift* dan *Flann Tree*.

DAFTAR PUSTAKA

- [1] Mikolajczyk, K., Schmid, C. (2005). "Performance Evaluation of Local Descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 27(10). pp. 1615-1630.
- [2] Calonder, M., Vincent Lepetit, Strecha, C., Pascal Fua. (2010). "BRIEF: Binary Robust Independent Elementary Features," *In ECCV*. pp. 778-792.
- [3] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. (2006). "Speeded-up Robust Feature," *Computer Vision and Image Understanding*. 110(3). 346-359.
- [4] Marius Muja, David G. Lowe. (2009). "FAST Approximate Nearest Neighbors with Automatic Algorithm Configuration," *In International Conference on Computer Vision Theory and Application (VISSAPP)*. pp. 331-340.
- [5] Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, Dieter Schmalstieg. (2010). "Real Time Detection and Tracking for Augmented reality on Mobile Device," *IEEE Transaction on Visualization and Computer Graphic*. Vol 16. No 3.
- [6] Mbsalam Ebrahimi, Walterio W. Mayol-Cuevas. (2011). "Adaptive Sampling for Feature Detection, Tracking and Recognition on Mobile Platforms," *IEEE Transactions on Circuits and Systems for Video Technology*.
- [7] Charles Norona, Tyagi, Vivek Kumar. (2010). "Non-distributed Object Recognition Potential on the Android Platform," *COT5930 – Digital Image Processing Fall*.
- [8] Niels Henze, Torben Schinke, Susanne Boll. (2009). "What is That? Object Recognition from Natural Features on a Mobile Device," *Research Paper University of Oldenburg*.
- [9] Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bismpiagiannis, Thanos Bismpiagiannis, Radek Grzeszczuk, Kari Pulli, dan Bernd Girod. (2008). "Outdoors Augmented Reality on Mobile Device Using Loxel-Based Visual Feature Organization," *Proc. of ACM international conference on multimedia information retrieval (ACM MIR)*. Vancouver. Canada.
- [10] Wei-Chao Chen, Yingen Xiong, Jiang Gao, Natasha Gelfand, Radek Grzeszczuk. (2007). "Efficient Extraction of Robust Image Features on Mobile Devices," *ISMAR*.
- [11] Ronald T. Azuma. (1997). "A Survey of Augmented Reality," *In Presence*:

- Teleoperators and Virtual Environments*. 6, 4. pp. 355-385.
- [12] Héctor López, Antonio Navarro, José Relañó. (2010). "An Analysis of Augmented Reality Systems," *Fifth International Multi-conference on Computing in the Global Information Technology*.
 - [13] J. Rekimoto, Y. Ayatsuka. (2000). "Cyber Code: Designing Augmented Reality Environments with Visual Tags," *Proc. Designing Augmented Reality Environments (DARE)*.
 - [14] Fiala, M. (2004). "ARTag Revision 1, A Fiducial Marker System Using Digital Techniques," *NRC/ERB-111*.
 - [15] Tom Drummond, Edward Rosten. (2006). "Machine Learning for High-Speed Corner Detection," *European Conf. Computer Vision (ECCV06)*. pp. 430-443.
 - [16] Christopher Evans. (2009). "Notes on the OpenSURF Library".
 - [17] Antonio Torralba, Fergus, R., Weiss, Y. (2008). "Small Codes and Large Image Databases for Recognition," *In Proc. of IEEE conference on computer vision and pattern recognition (CVPR)*. Anchorage. Alaska.
 - [18] Olsson, S. and Åkesson, P. (2009). "Distributed Mobile Computer Vision and Applications on the Android Platform," *Master's Thesis Paper. Lund University*.
 - [19] Tyagi, Vivek Kumar. (2010) "Object Recognition on the Android Platform Using Speeded Up Robust Features," *Master's Thesis Dissertation: Florida Atlantic University*.
 - [20] Sami Mohammad Halawani, Dzulkifli Mohammad. (2011) "Logo Matching Technique Based on Principle Component Analysis," *International Journal of Computer Vision and Applications*. Vol 1. No 1. April 2011

